



Runtime randomization and perturbation for virtual machines.

JAVIER CABRERA ARTEAGA

Licentiate Thesis in [Research Subject - as it is in your ISP]
School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden [2022]

TRITA-ICT XXXX:XX
ISBN XXX-XX-XXXX-XXX-X

KTH School of Information and
Communication Technology
SE-164 40 Kista
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägg-
es till offentlig granskning för avläggande av licentiatexamen i [ämne/subject]
[veckodag/weekday] den [dag/day] [månad/month] [år/2022] klockan [tid/time] i
[sal/hall], Electrum, Kungl Tekniska högskolan, Kistagången 16, Kista.

© Javier Cabrera Arteaga, [month] [2022]

Tryck: Universitetsservice US AB

Abstract

Write your abstract here...

Keywords: Keyword1, keyword2, ...

Sammanfattning

Write your Swedish summary (popular description) here...

Keywords: Keyword1, keyword2, ...

Acknowledgements

Write your professional acknowledgements here...

Acknowledgements are used to thank all persons who have helped in carrying out the research and to the research organizations/institutions and/or companies for funding the research.

Name Surname,
Place, Date

Contents

Contents	vi
List of Figures	vii
List of Tables	viii
List of Acronyms	ix
1 Introduction	1
1.1 Motivation	1
1.1.1 Why variants ?	1
1.1.2 Research questions	1
1.2 Contributions	1
2 Background and State of the art	3
2.1 CROW	3
3 Methodology	5
3.1 RQ1. To what extent can we artificially generate program variants for WebAssembly?	6
3.2 RQ2. To what extent are the generated variants dynamically different?	9
3.3 RQ3. To what extent do the artificial variants exhibit different exe- cution times on Edge-Cloud platforms?	12
4 Results	15
4.1 RQ1. To what extent can we artificially generate program variants for WebAssembly?	15
4.2 RQ2. To what extent are the generated variants dynamically different?	18
4.3 RQ3. To what extent do the artificial variants exhibit different exe- cution times on Edge-Cloud platforms?	21
5 Conclusions	25

List of Figures

3.1	The program variants generation for RQ1.	8
3.2	Dynamic analysis for RQ2.	10
3.3	Multivariant binary creation and workflow for RQ3 answering.	12
4.1	Pairwise comparison of programs' population traces in logarithmic scale. Each vertical group of blue dots represents a programs' population. Each dot represents a comparison between two program execution traces according to Metric 5.	19
4.2	Execution time distributions for Hilber_curve program and its variants. Baseline execution time mean is highlighted with the magenta horizontal line.	20
4.3	Execution time distributions. Each subplot represents the quantile-quantile plot of the two distributions, original and multivariant binary.	23

List of Tables

3.1	Corpora description. The table is composed by the name of the corpus, the number of modules, the number of functions, the lines of code range and the location of the corpus.	7
4.1	General program's populations statistics. The table is composed by the name of the corpus, the number of functions, the number of succesfully diversified functions, the cumulative number of generated variants and the cumulative number of unique variants.	16

List of Acronyms

Wasm
DTW

WebAssembly
Dynamic Time Warping

